Code Smell 61





SAP ABAP

Apex Apex

C C

C++

CloudFormation

COBOL COBOL

C# C#

CSS

**⋈** Flex

**GO** Go

5 HTML

🐇 Java

Js JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

**■** Text

TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



## Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

**∰** Bug

9

Security.allowDomain() should be used in a tightly focused man.  Vulnerability  The flash.system.Security.exactSetti property should never be set to factorize and the second security.  Vulnerability  Dynamic classes should not be used.  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowed.  Vulnerability  "default" clauses should be first	ngs alse used
be used in a tightly focused man  Vulnerability  The flash.system.Security.exactSetti property should never be set to f  Vulnerability  Dynamic classes should not be u  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowed.  Vulnerability	ngs alse used
The flash.system.Security.exactSetti property should never be set to factorial Vulnerability  Dynamic classes should not be used Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowed Vulnerability	alse used ne ctions
flash.system.Security.exactSetti property should never be set to f  Vulnerability  Dynamic classes should not be used to compare the configured to narrowly specify the domains with which local connection other Flex application are allowed.  Vulnerability	alse used ne ctions
Dynamic classes should not be use Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowed.  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowed.  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection are allowed.  Code Smell	ne ctions
"LocalConnection" should be configured to narrowly specify the domains with which local connect to other Flex application are allowable.  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection other Flex application are allowable.  Code Smell  "LocalConnection" should be configured to narrowly specify the domains with which local connection.	ne ctions
"LocalConnection" should be configured to narrowly specify the domains with which local connect to other Flex application are allowallow.  Compared to vulnerability	ctions
configured to narrowly specify the domains with which local connects other Flex application are allowallow.  Vulnerability	ctions
"default" clauses should be first	
	or last
☼ Code Smell	
Event types should be defined in metadata tags	
Code Smell	
Event names should not be hard in event listeners	coded
Code Smell	
The special "star" type should no used	t be
Code Smell	
Variables of the "Object" type shoot be used	ould
Code Smell	

Constant names should comply with a

All branches in a conditional structure should not have exactly the same

Classes that extend "Event" should

naming convention

Code Smell

implementation

📆 Bug

```
Tags
                                  Search by name...
Statements should end with
                                       Analyze your code
semicolons
                              convention
☼ Code Smell ☼ Minor ☒
In Flex, the semicolon is optional as a statement separator, but omitting
semicolons can be confusing.
Noncompliant Code Example
 function fun() {
   return // Noncompliant
         5 // Noncompliant
 print(fun()); // prints "undefined", not "5"
Compliant Solution
 function fun() {
   return 5;
 print(fun());
 Available In:
sonarcloud 🙆 | sonarqube
```

Security Hotspot (1)

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy

override "Event.clone()"  • Bug
Constructors should not dispatch events
Rug
"ManagedEvents" tags should have companion "Event" tags
<b>∰</b> Bug
Objects should not be instantiated inside a loop
Two branches in a conditional structure should not have exactly the same implementation
Constructor bodies should be as lightweight as possible
Only "while", "do" and "for" statements should be labelled
Statements, operators and keywords specific to ActionScript 2 should not be used
"for" loop stop conditions should be invariant
Unused function parameters should be removed
Code Smell